

FROM ZERO TO GAME DEV

*A Story of Two Friends, One Dream, and the Code That
Changed Everything*

Programming • Game Design • Game Development • Software Engineering

**FREE SAMPLE — INTRODUCTION + CHAPTER 1
PREVIEW**

*A complete journey from knowing nothing to building the games you always
dreamed of playing.*

Welcome to the Sample

Thank you for checking out From Zero to Game Dev. This sample includes the full Introduction — the story of Omar, the man whose life this book is built around — and the opening of Chapter One, where Omar's friend Alex begins teaching him programming from absolute scratch. The full book continues this conversation across four complete chapters: Programming, Game Design, Game Development, and Software Development, plus practical projects, code you can run yourself, and the rest of Omar and Alex's story.

Enjoying the sample? The full edition includes complete walkthroughs for building a text-based RPG, a 3D driving game in Unity, a Flask web backend with a live leaderboard, and a full design document template — all wrapped in the same friendly, funny, conversational style. Link to the full book is in the description / caption where you found this sample.

Introduction

The Man Who Had Everything Except a Direction

His name was Omar Khalil. Twenty-six years old, brown eyes that always looked slightly tired, and a habit of tapping his fingers on any flat surface when he was nervous — which was most of the time. He lived in a small apartment on the third floor of a crumbling building in Cairo, where the elevator had been broken since 2019 and the landlord had stopped pretending he would ever fix it. Omar climbed those stairs every single day, and every single day he stood at the top for a moment too long before going inside, as though he needed to prepare himself for the stillness of a life that had not moved in three years.

The apartment was clean — Omar was nothing if not tidy — but it felt hollow in the way that spaces feel hollow when the person inside them has not quite arrived yet. There was a second-hand couch the colour of old mustard, a desk covered in job rejection emails he had printed out and then decided not to read again, a mini-fridge that hummed like a nervous person, and a laptop so old it wheezed every time he opened more than two tabs. On the wall above the desk he had pinned a single yellow Post-it note that said, in his own handwriting: Something has to change. He had put it there three years ago. It had faded from yellow to the colour of old paper, but he had never taken it down.

Omar had graduated with a degree in Business Administration in 2021. He was not stupid — his professors had said so, sometimes even sounding like they meant it. But he was lost in the particular way that intelligent people get lost: he could see exactly how large the gap was between where he stood and where he wanted to be, and he had no map. Not even the beginning of one.

For two years after graduation he had tried. He had applied to marketing agencies, logistics companies, small banks, two startups that evaporated within months of hiring him, and once — in a moment of real desperation at eleven o'clock on a Tuesday night — a company that sold insurance to people who already had insurance. Nothing stuck. The interviews would go reasonably well and then go quiet. The phone never rang with good news. The emails came but they said the same polite thing in different fonts. He began to think there was something invisibly, fundamentally wrong with him, some factory defect that every interviewer could detect at a glance and that he would never be able to see in himself.

He had a box under his bed — an actual cardboard box, slightly damp at one corner from a leak he had asked the landlord about and been ignored — that contained the remnants of all the things he had tried and abandoned. A ukulele, bought during a phase when he was certain music was his calling and abandoned after three weeks when his fingers hurt too much. A stack of notebooks filled with business ideas, none of which had made it past the planning stage. A half-finished watercolour set. A language-learning app subscription he had cancelled after realising he was learning Italian for no reason except that he liked the sound of it. The box was not a record of failure, exactly. It was a record of a person searching, very hard, for something that kept not being in the places he looked.

What Omar Did Instead of Sleeping

The nights were the worst. During the day he could fill the hours with small intentional acts — a walk along the Nile where he would stand and watch the muddy green water and try to imagine it carrying something bad away with it, a coffee at the corner kiosk where the owner knew his order and never asked how the job search was going (a small mercy Omar had never properly thanked him for), a half-hearted update of his CV that he never quite finished. But at night, around eleven o'clock, when the city outside his window shifted from frantically loud to merely loud, Omar would sit at his wheezing laptop and play video games.

Not just play. Disappear. He would spend three, four, sometimes five hours inside worlds so carefully constructed and so much more interesting than his own that returning to the apartment afterwards felt like a kind of punishment — the lights too ordinary, the silence too unscored, the walls too close together and too unhelpful. He played GTA V and drove stolen cars through a glittering fake Los Angeles and felt, for a few hours, more alive than he had all day. He played Need for Speed and felt the wind of roads that had never been built, the speed of engines that did not exist. He played massive RPGs where he had quests with clear objectives and skills with visible levels and a progress bar — an actual, literal progress bar — that showed him at every moment exactly how far he had come and exactly how far he had left to go.

One night, at approximately two in the morning on a Wednesday in September, Omar stopped in the middle of a GTA mission and simply stared at the screen. The city was alive. The streetlights reflected on wet pavement in a way that was almost exactly right. A pedestrian walked past and complained, to no one

in particular, about rent. A pigeon landed on a trash can and regarded Omar with one sideways eye, with an expression that contained — and he knew this was absurd but felt it anyway — a personal judgment. And Omar thought, with a sudden clarity that surprised him: somebody built this. A real human being sat at a real desk and decided that this pigeon should exist. That it should be exactly here, on this can, at this angle, with this particular air of disdain.

He closed the game and opened a browser tab. He typed: How do people make video games? And for the first time in approximately two years, he was not tired.

That single search did not immediately solve anything. It led him to Reddit threads full of jargon he did not understand, to YouTube videos where men in dark rooms spoke about engines and scripts and sprites as though these were obvious words that every adult simply knew, to forum arguments between experienced developers about things so specific that Omar felt like he had wandered into a conversation that had been continuing for decades without him and with no intention of pausing to catch him up. He closed the laptop after an hour, more confused and simultaneously more interested than he had been when he started. And here is the thing that turned out to matter most: he came back the next night. And the night after that. Not with any plan. Just with the same question, approached from a slightly different angle. That was enough.

The Weekend That Changed the Coordinates

Three months after that first search, on a Tuesday morning in October with thin autumn sunlight coming through his one good window, Omar received a message from his childhood friend Alex. They had not spoken in almost a year. Alex had moved to a bigger city for a job that suited him — he was one of those people who had always seemed to know what the next step was before the current one had finished, which Omar had admired and slightly resented in equal measure. The message said: I heard you are still job hunting. Come visit this weekend. I have something that might actually be useful. Also I made too much biryani and need someone to help me deal with it.

The biryani detail was what decided it. Omar almost did not go — he had spent three years almost-not-doing things, and the habit had become deeply grooved. But a person who frames an invitation around surplus biryani is not performing helpfulness. They just genuinely want company and genuinely have something to say. He packed a bag on Friday evening, carried it down the broken-elevator stairs for what he did not yet know would be the last time as

the version of himself he had always been, and got on the bus.

Alex's apartment was not bigger or more expensive than Omar's, but it was alive in a specific way that Omar registered immediately without being able to name. Every surface had a purpose. There were two monitors on the desk — one showing code in a dark editor, one showing something that looked like a three-dimensional world in the process of being assembled. There were books that looked actually read, with cracked spines and sticky notes. There was a coffee machine that Alex operated with the focused attention of a scientist. And there was the biryani, as promised, filling the apartment with a warmth that made the whole place feel like somewhere things happened.

What followed that weekend became the education that no university had thought to give Omar. Alex was not a teacher and did not pretend to be one. He explained things the way you explain them to your closest friend: without performance, without condescension, without the fear that getting something slightly wrong would damage your authority. He sat next to Omar on the couch, opened the laptop, and said: You want to know how games are made. Let me show you how it actually works, from the very first thing to the last thing. And then he started.

This book is that conversation. Everything Alex told Omar that first weekend, and everything that came after in the months of calls and visits and late-night messages that followed, organised into four chapters. You are on that couch. The biryani is ready. Alex is about to open the laptop.

A note before we begin: You do not need a computer science degree. You do not need to be a maths genius. You need a working computer, curiosity that refuses to die, and the same quality Omar had the night he stared at that GTA pigeon and thought: somebody built this. Everything else is in the next four chapters.

CHAPTER ONE — PREVIEW

Programming: The Language of Worlds

In which Alex opens a laptop and Omar discovers that code is just giving very precise instructions to a very literal-minded assistant.

| Alex | Jordan |
|-----------------|--------------------|
| (o_o) | (^_^) |
| [laptop open] | [coffee in hand] |

[Saturday morning — coffee ready, laptop open, the whole weekend ahead of them]

[Alex's apartment, Saturday morning. The coffee is excellent. Omar is sitting on the couch looking at Alex's two monitors the way you look at something you know is important but cannot yet read. Which, in a way, is exactly what is happening.]

Alex:

Okay. Before we touch anything game-related — and I know you want to jump straight to making GTA — we need to talk about programming. Because everything in a game runs on code. The physics, the AI, the menus, the save system, the loading screen animation. All of it. Code.

Jordan:

When you say code — is it the green letters falling down a black screen? Because I want to be honest, that's my entire mental image.

[Alex looks at him for a long moment. Then laughs.]

Alex:

It's exactly that. You'll need a trench coat. No — it's just text. Organised text that a computer can understand and act on. That is genuinely all it is.

Jordan:

That's it?

Alex:

That's it. Every explosion, every rain effect, every NPC who panics when you fire a gun near them — someone typed text into a file and that text made it happen.

***** Hey — if you are reading this and thinking the same thing Jordan just thought — welcome. You are in exactly the right place. Most people give up right here, before they even start. You are not going to. *****

What Is Programming, Really?

Programming is writing instructions that a computer will follow exactly. Not approximately. Not with common sense. Exactly. This is the single most important thing to understand at the start: computers are extraordinarily powerful and extraordinarily literal. If you tell a computer to take three steps forward and there is a wall two steps ahead, the computer will walk into the wall. Repeatedly. Forever. Because you said three steps and the computer does not have opinions about walls you failed to mention.

This sounds frustrating. Sometimes it is. But it is also the secret advantage of programming: once you write an instruction correctly, the computer will follow it a million times without getting tired, without cutting corners, without deciding it has a better idea. The computer is the most perfectly obedient collaborator in history. It just needs you to speak its language.

[Alex pulls up a website — repl.it — a coding environment that runs directly in the browser. No installation required.]

Alex:

We're going to write your first line of code right now. Open that URL and type exactly this.

```
print('Hello, Omar. Welcome to the rest of your life.')
```

Jordan:

That's it?

Alex:

Hit Enter and find out.

[Omar types it. Hits Enter. The screen prints exactly the text between the apostrophes.]

Jordan:

It said it.

Alex:

It said it.

Jordan:

That's — okay that's actually really cool.

Alex:

Welcome to programming.

(The moment every programmer remembers. It feels ridiculous and magnificent at the same time. Both things at once.)

Python: The Best First Language

There are dozens of programming languages in common use, and they are good at different things. C++ powers most AAA games and runs extremely fast but is brutally difficult to learn. JavaScript powers the web. Swift powers iOS. For learning — for the first year, genuinely — Python is the right choice. It was designed to be readable. It looks almost like English. It does not demand that you understand a hundred preliminary concepts before you can do anything satisfying.

Alex:

Think of languages like actual languages. Spanish, Portuguese, and Italian are different, but if you speak Spanish fluently, the others come much faster because they share deep structure. Python is your Spanish. Once you know it, C# and JavaScript and C++ are much less frightening.

Jordan:

What does Python actually look like?

```
# Lines starting with # are comments. The computer ignores them.  
  
# They are notes for the human reading the code.  
  
player_name = 'Omar'  
  
player_score = 0  
  
lives_remaining = 3  
  
print('Welcome, ' + player_name)
```

```
print('Your score: ' + str(player_score))

print('Lives remaining: ' + str(lives_remaining))
```

Jordan:

I can read most of that. `player_name` equals Omar. `player_score` equals zero.

Alex:

That's all a variable is — a named box that holds a value. Give it a name, give it something to hold, and use that name anywhere in your code. The computer substitutes the value automatically.

Jordan:

So in a game, `health_points` equals 100.

Alex:

And when the player takes a hit: `health_points = health_points minus 25`. Now it's 75. Check if it's zero. If yes, show the death screen. That is literally, precisely how health systems work.

Jordan:

I thought there was more magic involved.

Alex:

The magic is that someone organised ten million of these tiny obvious steps into something that feels like driving a Lamborghini through a thunderstorm.

Jordan:

GTA reference. I appreciate the commitment.

This is where the sample ends. In the full book, Chapter One continues with variables and data types, control flow, loops, functions, a full introduction to Object-Oriented Programming, and a complete text-based dungeon game project you build from scratch with Omar and Alex. From there, Chapter Two covers game design (including a teardown of GTA and Need for Speed), Chapter Three walks you through Unity and builds a real driving game, and Chapter Four takes you from game development into web development, databases, Git, and shipping your first project to the world.

GET THE FULL BOOK

Ready to Keep Going?

The full edition of From Zero to Game Dev includes all four complete chapters, every code example fully working and ready to run, three hands-on projects (a text RPG, a 3D Unity driving game, and a Flask web leaderboard), full design document templates, and the rest of Omar and Alex's story all the way to the epilogue.

□ Get the full book — link in the description / caption where you found this sample.

From Zero to Game Dev • A Story of Two Friends

Programming • Game Design • Game Development • Software Development